

Spreadsheet Functions

Related Documentation

- [Overview of Spreadsheet Studies](#)
 - [Using Spreadsheet Studies](#)
 - [Spreadsheet Systems, Alerts and Automated Trading](#)
 - [Spreadsheet Study Inputs](#)
 - [Referencing Other Charts in Spreadsheet Study Formulas](#)
 - [Spreadsheet Studies Special Tasks](#)
 - [Sharing Your Spreadsheet Study With Another User](#)
 - [Working with Spreadsheets](#)
 - **Spreadsheet Functions**
 - [Spreadsheet Example Formulas and Usage](#)
-

On This Page

- [Introduction](#)
- [External Function Reference](#)
- [Using Spreadsheet Functions](#)
- [Function Parameter Delimiter](#)
- [Serial DateTime Values](#)
- [Using the Equivalent of COUNTIF, AVERAGEIF, MAXIF, MINIF, SUMIF](#)
- [Available Spreadsheet Functions](#)
 - [ABS](#)
 - [ACOS](#)
 - [ACOSH](#)
 - [ADDRESS](#)
 - [AND](#)
 - [ASIN](#)
 - [ASINH](#)
 - [ATAN](#)
 - [ATANH](#)
 - [AVEDEV](#)
 - [AVERAGE](#)
 - [AVERAGE_IGNOREZEROS](#)
 - [CEILING](#)
 - [CELL](#)
 - [CHOOSE](#)
 - [COLUMN](#)
 - [COLUMNS](#)

- [CONCATENATE](#)
- [CORREL](#)
- [COS](#)
- [COSH](#)
- [COUNT](#)
- [COUNTA](#)
- [COUNTBLANK](#)
- [CROSSFROMABOVE](#)
- [CROSSFROMBELOW](#)
- [CROSSOVER](#)
- [DATE](#)
- [DATEVALUE](#)
- [DAY](#)
- [DAYS360](#)
- [DEVSQ](#)
- [DEGREES](#)
- [EARLIESTNONZEROVALUE](#)
- [EDATE](#)
- [EOMONTH](#)
- [EVEN](#)
- [EXP](#)
- [FIND](#)
- [FISHER](#)
- [FISHERINV](#)
- [FLOOR](#)
- [FORECAST](#)
- [FRACTIME](#)
- [GetCorrespondingMatch](#)
- [HLOOKUP](#)
- [HOUR](#)
- [HULL_INTERMEDIATE](#)
- [HULL_RESULT](#)
- [IF](#)
- [INDEX](#)
- [INDIRECT](#)
- [INT](#)
- [INTDATE](#)
- [ISBLANK](#)
- [ISEMPTY](#)
- [ISERR](#)
- [ISEVEN](#)
- [ISLOGICAL](#)
- [ISNULL](#)
- [ISNUMBER](#)
- [ISODD](#)
- [ISRANGE](#)

- [ISREF](#)
- [ISSAMETIMETO HOUR](#)
- [ISSAMETIMETOMILLISECOND](#)
- [ISSAMETIMETOMINUTE](#)
- [ISSAMETIMETOSECOND](#)
- [ISTEXT](#)
- [LARGE](#)
- [LEN](#)
- [LEFT](#)
- [LN](#)
- [LOG](#)
- [LOG10](#)
- [MATCH](#)
- [MAX](#)
- [MAXL](#)
- [MEDIAN](#)
- [MID](#)
- [MILLISECOND](#)
- [MIN](#)
- [MING](#)
- [MINUTE](#)
- [MINZ](#)
- [MOD](#)
- [MODE](#)
- [MONTH](#)
- [MOSTRECENTNONZEROVALUE](#)
- [MROUND](#)
- [MROUNDDOWN](#)
- [MROUNDUP](#)
- [NETWORKDAYS](#)
- [NMATCH](#)
- [NORM.DIST](#)
- [NORM.S.DIST](#)
- [NOW](#)
- [ODD](#)
- [OFFSET](#)
- [OR](#)
- [PEARSON](#)
- [PERCENTILE](#)
- [PERCENTRANK](#)
- [PROB](#)
- [RADIANS](#)
- [RAND](#)
- [RANDINT](#)
- [RIGHT](#)
- [ROUND](#)

- [ROUNDDOWN](#)
- [ROUNDUP](#)
- [ROW](#)
- [ROWS](#)
- [SEARCH](#)
- [SECOND](#)
- [SIGN](#)
- [SIN](#)
- [SINH](#)
- [SLOPE](#)
- [SMALL](#)
- [SQRT](#)
- [STANDARDIZE](#)
- [STDEV](#)
- [STDEV.P](#)
- [STDEV.S](#)
- [SUM](#)
- [SUMPRODUCT](#)
- [TAN](#)
- [TANH](#)
- [TEXT](#)
- [TIME](#)
- [TIMEVALUE](#)
- [TODAY](#)
- [TRIMMEAN](#)
- [TRUNC](#)
- [TRUNC HOUR](#)
- [TRUNC MIN](#)
- [TRUNC SEC](#)
- [TYPE](#)
- [VALUE](#)
- [VLOOKUP](#)
- [WEEKDAY](#)
- [WEEKNUM](#)
- [WEIGHTEDMOVINGAVERAGE](#)
- [WORKDAY](#)
- [YEAR](#)
- [ZTEST](#)

Introduction

This documentation page documents all of the available Spreadsheet functions.

Spreadsheet functions are entered into cells. They take parameters and data, make calculations on the parameters and data, then return the result to the same cell they are entered in.

External Function Reference

For more detailed information about any of the Spreadsheet functions documented on this page, refer to the [Microsoft Excel Function Reference](#).

Using Spreadsheet Functions

To use a Spreadsheet function, use the = sign to indicate a formula, enter the function name, then enter the parameters and data enclosed in parentheses (). Separate each parameter with a comma ",". The formula does not only have to contain a single function, it can contain anything that can be part of a formula as described on the [Working with Spreadsheets](#) documentation page.

Spreadsheet function parameters can be Numbers, Cell and Range References, Other Functions, Text Strings (the text needs to be in quotation marks), Dates and Times, Logical Values (TRUE, FALSE), and Error Values.

Any of these types can be used as a parameter as long as they are compatible with or can be converted to the parameter type specified in the function format in the list of Spreadsheet Functions. Parameters can also use operators. Use a cell or range reference as a parameter or as part of an expression, if you want to use data for a function parameter that is located in a cell or a range.

Examples:

`=SQRT(4)` This function can be entered in any cell and will return 2.

`=SQRT(B1)` Returns the square root of the value in cell B1.

`=SQRT(4+B1)` Returns the square root of 4 plus the value in cell B1. The parameter to this function is called an Expression.

`=SUM(F4:F24)` Returns the sum of cells F4 through F24.

The format for some functions specify an Expression. An expression is just like a formula. Except that it is not prefixed with an =. Therefore, anything you can use in a formula can be used in an expression. See the [Working with Spreadsheets](#) documentation page for information on formulas and expressions. The simplest form of an expression is a single number (Example: 5). A logical expression is one that returns TRUE if it evaluates to a nonzero number and returns FALSE if it evaluates to 0.

If the function format shows a parameter in brackets ([]), then that parameter is optional.

When a parameter specifies a list, then use a reference to a range or a comma separated list.

A Text String is text and can be a single character or a series of characters enclosed in quotation ("") marks.

In the function declarations below, the parameter name **Value** means any type of numeric value, text or a cell reference.

Function Parameter Delimiter

The delimiter to use between parameters specified in Spreadsheet functions can either be a comma (,) or a semicolon (;). Example using comma delimiter: `=SUM(1, 2)`. Example using semicolon delimiter: `=SUM(1,5 ; 10)`.

Which delimiter is used depends upon the **Region** setting in

Global Settings >> Spreadsheet Settings. If the region you have set uses a comma (,) as a decimal point delimiter, then the Spreadsheet function parameter delimiter will be a semicolon (;). Keep this in mind when reviewing the functions below. The functions below use a comma (,) delimiter between parameters.

Defining a Range of Data

Some Spreadsheet functions can take a range of data as a parameter. This is specified when two cell references are separated by a colon (:). For these functions, the first cell reference (before the colon) specifies the upper left-hand corner of the range and the second cell reference (after the colon) specifies the lower right-hand corner of the range. This conceptual rectangle must always be drawn from upper left to lower right. As such, the row number specified by the second cell reference must always be equal to or greater than the row number specified by the first cell reference.

Specifying a range of data within a single column is well understood, as the range is easily viewable and creates a single column of data. It is, however, still necessary that the row values are increasing. For example, specifying the range **B3:B5** is valid, but **B5:B3** is not and will give a #REF error.

It is also possible to specify a range of data that crosses between columns. When doing this, the data that is contained within the rectangle formed from the first cell reference to the second cell reference is the range of data that will be used. For example, specifying a range of **B3:C5** will include the data in the following cells: B3, B4, B5, C3, C4, and C5. As noted above, giving a range of values where the row number of the second cell reference is less than the row of the first cell reference will result in a #REF error.

Chartbook1						
Sheet3						
C5	A	B	C	D	E	F
1	[ID0] NQH19 [M...			Formula Source ...		
2	Date Time	[ID0.SG1] Open	[ID0.SG2] High	[ID0.SG3] Low	[ID0.SG4] Last	[ID0.SG5] Volume
3	2018-12-19 15:...	6334.00	6335.75	6333.75	6335.00	37
4	2018-12-19 15:...	6335.75	6335.75	6333.25	6334.00	56
5	2018-12-19 15:...	6335.50	6337.00	6333.25	6336.00	137
6	2018-12-19 15:...	6335.25	6336.25	6333.00	6335.00	163
7	2018-12-19 15:...	6330.50	6337.50	6330.50	6334.75	325
8	2018-12-19 15:...	6331.00	6332.00	6329.00	6330.50	240
9	2018-12-19 15:...	6333.00	6335.50	6328.00	6330.50	474
10	2018-12-19 15:...	6343.50	6343.50	6343.50	6343.50	379
11	2018-12-19 15:...	6348.00	6348.00	6348.00	6348.00	247
12	2018-12-19 15:...	6352.25	6352.25	6352.25	6352.25	244
13	2018-12-19 15:...	6354.00	6354.00	6354.00	6354.00	338
14	2018-12-19 15:...	6358.75	6358.75	6358.75	6358.75	1237
15	2018-12-19 15:...	6352.75	6359.00	6351.50	6358.50	502
16	2018-12-19 15:...	6352.00	6356.75	6348.75	6352.75	678
17	2018-12-19 15:...	6347.75	6352.50	6347.00	6352.00	376
18	2018-12-19 15:...	6349.50	6351.25	6346.25	6348.00	494
19	2018-12-19 15:...	6353.25	6355.00	6349.00	6349.00	472
20	2018-12-19 15:...	6358.25	6358.75	6353.25	6353.50	378

Specifying a range of B3:C5 creates a virtual rectangle that gives the data in cells B3, B4, B5, C3, C4, and C5.

Serial DateTime Values

Sierra Chart Spreadsheets store Dates and Times as double precision floating point numbers which represent the time since 1899-Dec-30 at 00:00:00. This date-time is not in any particular time zone. It can represent any time zone. To set the time zone, refer to [Time Zone](#).

The integer part of the floating-point number represents the days and the fractional part, to the right of the decimal place, represents the time. This is exactly the same way as Excel and OpenOffice Calc represent date and Time values. This is called a Serial Date Time value.

Note that due to historical issues Excel does have a mistake in that it includes February 29, 1900, which did not exist. Therefore, Excel has 1899-12-31 as day 0 and there will be a 1 day discrepancy between Sierra Chart and Excel for dates prior to March 1, 1900.

The **Spreadsheet Study**, **Spreadsheet System/Alert** and the **Spreadsheet System for Trading** studies output Date-Time values to column **A** using this format.

Time Examples: 12 PM would be represented as .5. 1 minute or 00:01:00 would be represented as 1.0/1440.0. There are 1440 minutes in a day.

00:01:10 would be represented as 70.0/86400.0. There are 86,400 seconds in a day. 1 second evaluates to 1.15740740E-5.

Date Examples: 1900-Jan-2 would be represented as 3.

Comparing Serial Date-Time Values

Since Date-Time values are stored as floating-point numbers, they are imprecise when you are performing comparisons to them. Two Date-Time values that are the same when they are formatted as a Date and Time string, may not give you an exact comparison due to floating point error. You can see the exact values if you format the spreadsheet cell containing the Date-Time value to a number with 9 decimal places.

A solution when doing an equals comparison between 2 Date-Time values between two different sheets in the Spreadsheet, is to use a formula similar to the following:

`=ROUND(A3,8)=ROUND(Sheet2!A3,8)` .

Sierra Chart internally stores Date-Time values in the same way as Spreadsheets do. For further information, refer to the [SCDateTime](#) data type page.

Any of the available [Spreadsheet Functions](#) which accept a Serial DateTime Value parameter or return a Serial DateTime Value can be used when working with Date-Time values in a Spreadsheet.

Using the Equivalent of COUNTIF, AVERAGEIF, MAXIF, MINIF, SUMIF

The Spreadsheets in Sierra Chart do not support the following functions: **COUNTIF, AVERAGEIF, MAXIF, MINIF, SUMIF**.

These are not supported due to the complexity of supporting criteria text. However, there is an alternative method to effectively perform these functions.

The [Spreadsheet Studies](#) support up to 60 formula columns. It is possible to use additional formula columns with the basic supported Spreadsheet functions to accomplish the same result as these unsupported functions.

The formula `=COUNTIF(AA3:AA12,">50")` can be implemented in the Sierra Chart Spreadsheets as follows:

In one of the available Spreadsheet formula columns, by default K through Z, enter this formula:

`=IF(AA3>50,1,0)`. Assuming the prior formula was entered in formula column X, then enter

`=SUM(X3:X12)` in another formula column. The result of this last formula will be the same as

`=COUNTIF(AA3:AA12,">50")`.

Available Functions

Format	Description

ABS (Number) (Link)	<p>The absolute value of Number. If the given Number is an integer, the return value will be an integer. If Number value is a decimal, the return value will be a decimal. Returns #VALUE! if Number is not a number.</p>
ACOS (Number) (Link)	<p>The arc cosine of Number, in radians. Returns #NUM! if Number is outside the range [-1,1]. Returns #VALUE! if given Number is not a number.</p>
ACOSH (Number) (Link)	<p>Returns the inverse hyperbolic cosine of the given Number. Returns #NUM! if Number is less than 1. Returns #VALUE! if the given Number is not a number.</p>
ADDRESS (row, column, [ref_type], [ref_style], [sheet_name]) (Link)	<p>Returns an address as a text string. Return value is based on row or column are less than or equal to 255.</p> <p>ref_type:</p> <ul style="list-style-type: none"> 1 = Absolute Row and Column. 2 = Absolute Row and Relative Column. 3 = Relative Row and Column. 4 = Relative Row and Absolute Column.

<p>AND(Boolean, [...]) (Link)</p>	<p>Returns TRUE if and only if all the given parameters evaluate to TRUE. Otherwise, returns FALSE. Returns #VALUE! if any of the given parameters cannot be interpreted as a Boolean value. Examples:</p> <p>=AND(E3 > 10, A3 < 5) (Spreadsheet Study form)</p> <p>=AND(C > 100, S < 50) (Simple Alert formula)</p> <p>=OR(AND(H > 100, S < 50), AND(L < 80, S < 50)) (Simple Alert formula)</p>
<p>ASIN(Number) (Link)</p>	<p>Returns the arc sine of the given Number, in radians. Returns #NUM! if the given Number is outside the range [-1, 1]. Returns #VALUE! if the given Number is not a number.</p>
<p>ASINH(Number) (Link)</p>	<p>Returns the inverse hyperbolic sine of the given Number. Returns #VALUE! if the given Number is not a number.</p>
<p>ATAN(Number) (Link)</p>	<p>Returns the arc tangent of the given Number, in radians. Returns #VALUE! if the given Number is not a number.</p>

ATANH (Number) (Link)	Returns the inverse tangent of the give Returns #NUM! if Number greater or e or less or equal that #VALUE! if the given not a number.
AVEDEV (Number, [...]) (Link)	Returns the avera absolute deviations numbers from th Returns #VALUE! if r are found.
AVERAGE (Number, [...]) (Link)	The average of all c Numbers. Null valu counted as part of t Returns #NUM! if al Numbers are r #VALUE! if one of th given could not be in a number.
AVERAGE_IGNOREZEROS (Number, [...]) (Link)	The average of all c Numbers, except fo that are equal to zero are not counted as average. Returns #Nl given Numbers are e zero. Returns #VALU the Numbers given c interpreted as a numk

CEILING(Number, [Multiple = 1]) ([Link](#))

Rounds the given Number to the next number that is a multiple of the given Multiple, if the given Number does not already satisfy this condition. If Multiple is not given, the given Number is rounded up to the next integer. If Multiple is a non-integer, the returned value is rounded to the nearest integer, otherwise it is a double. Returns #N/A if the given Multiple is zero. Returns #VALUE! if the given Number or Multiple is not a number.

CELL(Text, [ReferenceOrRange]) ([Link](#))

Returns information about a cell. If the first argument is "row" the column or row number is returned. If the first argument is "contents" the content of the cell is returned, if it is "type" the cell type will be returned. If the cell is empty, "I" for empty cell, "L" for constant text and "N" for values). If the second argument is provided it is either a reference to return information about the referenced range, or a single cell to return information about the first cell of a range.

CHOOSE(ValueNumber, Value1, [Value2, [...]]) [\(Link\)](#)

Returns one of the arguments based on ValueNumber. If ValueNumber is 1, the first argument is returned. If a cell range reference is given, each of the cells in the range are treated as values for this function. For example, CHOOSE(3, A1:A3) returns the value of the third cell in the range A1:A3. Returns #REF! if the given ValueNumber is outside of the range of values given. Returns the given ValueNumber if no range is given.

COLUMN([Reference]) [\(Link\)](#)

Returns the absolute (not relative) number of the column of the cell Reference. If Reference is given, then the number of the column containing the cell Reference is returned. If Reference is given as a cell reference value, the absolute number of the column (A) is 1.

COLUMNS(Range) [\(Link\)](#)

Returns the number of columns in the given Range. Returns #VALUE! if the given Range is not a cell range reference type.

CONCATENATE (Text, [...]) (Link)	Combines the given values in the order that they are given and returns the combined text as a single text value. Returns #VALUE! if one of the parameters could not be interpreted as a text value.
CORREL () (Link)	See: PEARSON
COS (Number) (Link)	The cosine of the given number. Returns #VALUE! if the number is not a number.
COSH (Number) (Link)	Returns the hyperbolic cosine of the given Number. Returns #VALUE! if the given Number is not a number.
COUNT ([Values, [...]]) (Link)	Returns the number of values given in the list. If a range is given, this counts the number of cells in the range. Numeric values that have a value of zero, integer or double.
COUNTA ([Values, [...]]) (Link)	Returns the number of non-empty values given in the parameters list. If a range is given in the list, this counts the non-empty values in the range.

<div>COUNTBLANK([Values, [...]]) (Link)</div>	Returns the number of empty/blank values in the parameter list. If a range is in the list, this counts the values in the range.										
<div><div>CROSSFROMABOVE(range1, range2)</div><div>Example: =CROSSFROMABOVE(AA3:AA4,AB3:AB4)</div><div>Example: =CROSSFROMABOVE(E3:E5, AA3:AA5)</div><div>Example: =CROSSFROMABOVE(ID1.SG1@3:ID1.SG1@5, ID2.SG1@3:ID2.SG1@5)</div><table><tr><td>Study</td><td>Alert</td><td>Condition</td><td>Formula</td><td>Example:</td></tr><tr><td></td><td></td><td></td><td>CROSSFROMABOVE(SG1, SG2) (Link)</td><td></td></tr></table></div>	Study	Alert	Condition	Formula	Example:				CROSSFROMABOVE(SG1, SG2) (Link)		<div>Compares 2 ranges. Each range needs to have at least 2 numbers and 3 numbers. For the accuracy use a range that includes 3 values. This function determines if the first range of values crosses the second range from above. Returns a boolean value.</div> <div>TRUE = The first range crosses the second Range from above.</div> <div>FALSE = The first range does not cross the second range from above.</div> <div>If one of the ranges is a constant, the crossover needs a constant, then it needs a study Subgraph column which contains a constant value.</div>
Study	Alert	Condition	Formula	Example:							
			CROSSFROMABOVE(SG1, SG2) (Link)								

CROSSFROMBELOW(range1, range2)

Example: **=CROSSFROMBELOW**(AA3:AA4,AB3:AB4)

Example: **=CROSSFROMBELOW**(E3:E5, AA3:AA5)

Example: **=CROSSFROMBELOW**(ID1.SG1@3:ID1.SG1@5,
ID2.SG1@3:ID2.SG1@5)

Study	Alert	Condition	Formula	Example:
			CROSSFROMBELOW (SG1, SG2)	(Link)

Compares 2 ranges
Each range needs to
least 2 numbers and
3 numbers. For the
accuracy use a range
includes 3 values
function. Determines
range of values in
second range from
Returns a boolean value
TRUE = The first range
the second Range from
FALSE = The first
not cross the second
below.

If one of the ranges
the crossover needs
constant, then it needs
a study Subgraph
column which contains
constant value.

CROSSOVER(range1, range2)

Example: **=CROSSOVER(AA3:AA4, AB3:AB4)**

E x a m p l e : **=CROSSOVER(ID0.SG4@3:ID0.SG4@5, ID1.SG1@3:ID1.SG1@5)**

Study Alert Condition Formula Example: **CROSSOVER(SG1, SG2)**

These study Subgraph identifiers are automatically interpreted as ranges.

Study Alert Condition Formula example with multiple instances:

=OR(CROSSOVER(C, -500), CROSSOVER(C, -1000), CROSSOVER(C, 500), CROSSOVER(C, 1000)) [\(Link\)](#)

This function compares two ranges of values. Each range must contain at least 2 numbers. The function can contain 3 numbers for the greatest accuracy, which includes 3 values. Returns a value which indicates the type of crossover:
1 = The first Range is greater than the second Range from both sides
-1 = The first Range is less than the second Range from both sides
0 = The ranges do not cross each other.

If one of the ranges is a constant, then the crossover needs a constant, then it needs a study Subgraph column which contains a constant value.

DATE(Year, Month, Day) [\(Link\)](#)

Returns a [Serial Date](#) for the given Year, Month, and Day. Returns #NUM! if Year, Month, and Day are not valid. Specify a valid date. #VALUE! if any parameters could not be interpreted as integer.

<p>DATEVALUE(Text) (Link)</p>	<p>Returns a Serial Date for the given Text, into a date string. The Text given in the form including quotes, where m is numerical month (one or two digits), d is numerical day (one or two digits) and YYYY is year (four digits) - For example '12/31/2012'. Returns #NUM! if the given Text cannot be interpreted as a date. Returns #VALUE! if the given Text cannot be interpreted as text value.</p>
<p>DAY(Serial DateTime Value) (Link)</p>	<p>Returns the day of the month for the given serial date. 1 is returned for the first day of the month. Returns the given DateTime interpreted as a serial date.</p>
<p>DAYS360() (Link)</p>	<p>Returns the number of days between the start date and end date using a 360 day month calendar used in accounting systems.</p>
<p>DEVSQ(Numbers, [...]) (Link)</p>	<p>Returns the sum of the squares of deviations from the mean. Returns #VALUE! if any arguments are found.</p>
<p>DEGREES(Radians) (Link)</p>	<p>Converts the given Radians to degrees. Returns #VALUE! if Radians is not a number.</p>

EARLIESTNONZEROVALUE (Range) (Link)	Returns the last non zero value in the given Range. Values that are considered equal to zero. Returns a null value if there are no non zero values in the given range. Either null or zero.
EDATE (Date, Months) (Link)	Returns the serial date value of the date given number of Months specified Date. If Months is greater than 0, the return value is the date that is the indicated number of months before the specified date. If the same day of the month does not exist in the resulting month, then the last day of the month is returned. Returns #VALUE! if the arguments are not integers.
EOMONTH (Date, Months) (Link)	Returns the serial date value of the last day of the month that is the number of months specified date. If the number of months argument is less than 0, the return value is the date of the last day of the month indicated number of months before the specified date. Returns #VALUE! if the arguments are not integers.

EVEN(Number) ([Link](#))

If the number is not even, it is rounded up to the next even number. If the Number is less than 0, it is rounded away from 0. #VALUE! if the argument is not a number.

EXP(Number) ([Link](#))

Returns the value of e raised to the power of the given Number. #VALUE! if the given argument is not a number.

FIND(SubString, FullString, [StartPosition = 1]) [\(Link\)](#)

Returns the starting position of the first instance of SubString text within FullString text. The StartPosition parameter is 1-based, meaning the first position where SubString is found. If StartPosition is not specified, the return value will be the position of the first instance of SubString at the beginning of the given FullString. By default, FIND begins its search at the beginning of the given FullString for the given SubString. If StartPosition is specified, but this can be overridden by specifying a value greater than 1 for the optional StartPosition parameter. The search is case-sensitive (use SEARCH for case-insensitive search). Returns 1 if the given SubString is found within the given FullString. Returns #VALUE! if the given SubString is not found, or the given StartPosition is beyond the length of the given FullString, or one of the arguments is not a text value, or the given StartPosition is not a numeric value. Returns #NUM! if the given StartPosition is less than 1.

FISHER(Number) [\(Link\)](#)

Returns the Fisher Transformation for the given number. Returns #NUM! if the given Number is less than or equal to -1, or greater than or equal to 1. Returns #VALUE! if the given Number is not a number.

FISHERINV (Number) (Link)	Returns the Inverse Transformation for number. Returns #VALUE! if argument is not a number.
FLOOR (Number, [Multiple = 1]) (Link)	Rounds the given Number down to the next number multiple of the given Multiple. If the given Number already satisfies this Multiple is 1 or not, the given Number is returned. If the given Number is rounded to the next whole number, the returned value will be the given Number. If the given Multiple is an integer, the returned value will be the given Number. Otherwise, it will be the given Number rounded down to the next whole number. Returns #NUM! if Multiple is zero or negative. Returns #VALUE! if given Number or Multiple is not a number.
FORECAST (X, KnownYs, KnownXs) (Link)	Returns a predicted linear regression based on known X and known Y values. Refer to Using the FORECAST Function for more information.
FRACTIME (DateTime) (Link)	Returns the fractional part of the given DateTime. Returns 0 if the given value is empty. Returns the given DateTime interpreted as a serial value.
	Searches a range of values for

GetCorrespondingMatch(SearchColumn,
SearchValue, NearestOrExact,
SearchRangeOrderingFlag,
SameValueResolution, ReturnResultColumn,
ReturnResultRowOffset) ([Link](#))

and returns a reference to the cell at the same found row index, another column range.

SearchColumn: The column searched. Example: AA3:AA102

SearchValue: The value to search for in SearchColumn. Only doubles and text values are supported. If a numeric value returns an error. Searching for a text value is supported. Any value given for SearchValue cannot be interpreted as a double. If a TRUE/FALSE value results in an error. Example: 50.0.

NearestOrExact: 0 = exact match. While iterating through elements, and an exact match is found, the NearestOrExact parameter and the NearestOrExact parameter. The nearest match will instead be returned. Example: 0.

SearchRangeOrderingFlag: 1 = SearchColumn values are ascending. (Lowest row index has the lowest numbers. Highest row index has the highest numbers).

0 = SearchColumn is unordered. The first match will be used. Using a value of 0 may result in lower performance. So it is best to use 0 or -1 if possible.

-1 = SearchColumn values are descending. (Lowest numbered row has the highest value. Highest numbered row has the lowest value).

Example: 0.

SameValueResolution: When there are multiple values which repeat within an ordered range, this parameter indicates how to resolve the match. Use higher numbered row index or lower numbered row index. Example: 0

ReturnResultColumn: The column to return. A reference will be returned for the first row where the match was found.

can be the same as **SearchC**
AA3:AA102.

ReturnResultRowOffset: A po
row offset for the row
ReturnResultColumn. Example:

E x a m p
GetCorrespondingMatch(AA\$3:A
0, AA\$3:AA\$402, 0)

Searches for the giv
the top-most row o
Range, and returns
the the cell at the s
of the found value, a
the given **RowInRang**

RowInRange is a
within the given **Rang**
is the top-most row o
Range.

I f **ApproximateMatc**
as **TRUE**, and no exa
found, the last value
given **Range** that is le
given **Value** will be us

W h e n **Approximat**
given as **TRUE**, the v
top-most row of the g
are assumed to be in
order.

I f **ApproximateMat**
given or given as **FAI**
given **Value** is not f
is returned.

Returns **#VALUE!** if

HLOOKUP(Value, Range, RowInRange,
[ApproximateMatch]) [\(Link\)](#)

	<p>Value cannot be resolved to an actual value (such as a cell reference is given), or RowInRange cannot be interpreted as an integer or the given Approximate cannot be interpreted as a boolean value.</p> <p>Returns #REF! if Range is not valid, or RowInRange is beyond the number of rows in Range.</p> <p>Returns #NUM! if RowInRange is less than 1.</p>
<p>HOUR(Serial DateTime Value) (Link)</p>	<p>Returns the hour for the given Serial DateTime Value. The function will return a value in the range 0-23. Returns the given DateTime interpreted as a serial value.</p>

HULL_INTERMEDIATE(Reference, Value)
[\(Link\)](#)

Returns the calculation values for the Hull Moving Average. This function can be used in combination with the HULL_RESULT function. In this function, the first parameter is the cell reference to the first (top) cell of the input range, and the **Value** parameter is the value defined in the Moving Average function. The second parameter is the number of cells to include in the calculation. The **Value** parameter is used in this function to match the value of the HULL_RESULT function. If the number of cells is 0, the value of 0 is returned. Returns the first parameter if the second parameter is not an integer, or if any of the values within the input range is not a number. Returns #N/A if the Value parameter is not enough for a calculation.

HULL_RESULT(Reference, Value) [\(Link\)](#)

Returns the final values for the H Average. This is used in combination with this function in and the HULL_INTERMEDIATE function in another cell. **Reference** is to the first cell of the HULL_INTERMEDIATE column and the **Value** is the length of the Moving Average (the number of cells). The **Value** used in this function needs to match the value used in the HULL_INTERMEDIATE function. Empty cells are considered to have a value of 0. Returns #VALUE! if the first parameter is not a reference, the second parameter is not an integer, or if the values within the input range are not a number. Returns the Value parameter if not enough data for a calculation.

Returns the value of the given Condition. Returns TRUE. Returns the FalseValue if the given Condition is equal to FALSE. Returns #VALUE! if the Condition cannot be interpreted as a value.

IF(Condition, TrueValue, FalseValue) [\(Link\)](#)

Multiple IF statements can be strung together to create a nested IF/ELSE IF function. To create a nested IF function, simply put additional IF statements for the arguments. For example, **IF(A > B, 1, IF(B > C, 2, 0))** will evaluate as: if A is greater than B, then 1, else if B is greater than C, then 2, else 0.

INDEX(Range, Index)
Row, Column) [\(Link\)](#) or **INDEX**(Range,

Returns a cell reference from the given Range, either the given Index or the given Column within the Range using Index (only one argument after Range) on a list of multiple rows and columns. Items are ordered down rows first, and then across columns.

Index, Row, and Column are 1-based, meaning use of 1 will return the first item. Returns #REF! if the given Row, or Column is outside the given Range. Returns the given Range if the reference range type is an array. Given Index, Row, or Column could not be integer values.

INDIRECT(ReferenceText) [\(Link\)](#)

Returns a reference to the given ReferenceText. ReferenceText must be a value that can be parsed as a standard reference in formulas. This function supports basic cell reference, basic cell range reference, basic column range reference, and advanced reference supported. Returns a #VALUE! error if the given ReferenceText cannot be parsed or is invalid. Returns #VALUE! if given ReferenceText is not a value.

An example ReferenceText is "E3". ReferenceText created by a formula. For example, if cell H5 contains the number 3, then the formula =**CONCATENATE**("E",

INT(Number) [\(Link\)](#)

Rounds the given Number down to the next whole number. Returns #VALUE! if Number is not a number. Returns 0 if Number is an empty cell or like a reference to a cell that contains no data.

INTDATE (DateTime) (Link)	Returns the integer part of the given DateTime value. Returns 0 if the given value is empty. Returns the given DateTime value interpreted as a serial date value.
ISBLANK (Value) (Link)	Returns TRUE if the given Value is blank, which means it is an empty cell. Otherwise returns FALSE.
ISEMPTY (Value) (Link)	Returns TRUE if the given Value is empty, which means it is an empty cell. Otherwise returns FALSE.
ISERR (Value) (Link)	Returns TRUE if the given Value is an error value type (e.g., #N/A, #DIV/0!, #VALUE!, #REF!, #NAME?, #NUM!). Otherwise returns FALSE. IFERROR(A, B, C) can be written as IF(ERRO(A), B, C) . IFNA(A, B, C) can be written as IF(A=#N/A, B, C) .
ISEVEN (Number) (Link)	Returns TRUE if the given Number is an even integer. Otherwise returns FALSE. If the given Number is a decimal, only the integer part is used. Returns #VALUE! if the given Number is not a number.

ISLOGICAL (Value) (Link)	Returns TRUE if the given Value is a boolean value. Otherwise returns FALSE.
ISNULL (Number) (Link)	Returns TRUE if the given Number is null, which means it has no value. Otherwise returns FALSE.
ISNUMBER (Number) (Link)	Returns TRUE if the given Number is an integer or decimal number. Otherwise returns FALSE.
ISODD (Number) (Link)	Returns TRUE if the given Number is an odd integer. Otherwise returns FALSE. If the given Number is a decimal number, only the integer part is used. Returns #VALUE! if the given Number is not a number.
ISRANGE (Value) (Link)	Returns TRUE if the given Value is a range reference. Otherwise returns FALSE.
ISREF (Value) (Link)	Returns TRUE if the given Value is a reference. Otherwise returns FALSE.

ISSAMETIMETOHOUR(DateTime1,
DateTime2) ([Link](#))

Returns TRUE if the for both of the given [time values](#) are at the down to the hour. Example: comparing 12:00:00 to 12:59:59.999 would return TRUE, but 12:59:59.999 to 1 would return FALSE. The hour component of both values is ignored. Returns #VALUE! if either of the given values cannot be interpreted as a time value.

ISSAMETIMETOMILLISECOND(DateTime1,
DateTime2) ([Link](#))

Returns TRUE if the for both of the given [time values](#) are at the down to the millisecond. Example: comparing 12:00:00.000 to 12:00:00.001 would return TRUE, but 12:00:00.000 to 12:00:00.000 would return FALSE. The date component of both values is ignored. Returns #VALUE! if either of the given values cannot be interpreted as a serial date-time value.

<p>ISSAMETIMETOMINUTE(DateTime1, DateTime2) (Link)</p>	<p>Returns TRUE if the for both of the given time values are at the down to the minut comparing 12:00:C 12:00:59.999 wou TRUE, but 12:00:59.999 to 1 would return FALSE component of both ignored. Returns # either of the given va be interpreted as a time value.</p>
<p>ISSAMETIMETOSECOND(DateTime1, DateTime2) (Link)</p>	<p>Returns TRUE if the for both of the given time values are at the down to the secon comparing 12:00:C 12:00:00.999 wou TRUE, but 12:00:00.999 to 1 would return FALSE component of both ignored. Returns # either of the given va be interpreted as a time value.</p>
<p>ISTEXT(Value) (Link)</p>	<p>Returns TRUE if the is a text value type returns FALSE.</p>

<p>LARGE(Numbers, NthLargest) (Link)</p>	<p>Returns the Nth largest from an array of numbers. The first argument is the array, the last argument is the number of the largest number. If the number is 1, it means the largest number, 2 means the second largest and so on.</p> <p>Empty cells are not counted. If the last argument is TRUE, it will be considered as 1. If the last argument is FALSE, it will be considered as 0. Returns #VALUE! if the last argument is less than 1 or greater than the number of numerical values (excluding empty cells) in the array.</p>
<p>LEN(Text) (Link)</p>	<p>Returns the number of characters in the string. Returns #VALUE! if the argument is not a string.</p>
<p>LEFT(Text, Count) (Link)</p>	<p>Returns the given Count of characters from the left of the given Text string. If Count is greater than the length of the given Text string, the entire string will be returned. If Count is 0, an empty string is returned. If the given Count is negative, the entire length of the string, except for the first Count characters, will be returned. Returns #VALUE! if Text is not a string, or Count is not a number.</p>

LN (Number) (Link)	Returns the natural logarithm (using base e) of the given Number. Returns #NUM! if the given Number is less than or equal to 0. Returns the given Number if the Number is 1.
LOG (Number, [Base = 10]) (Link)	Returns the logarithm of the given Number using the given Base. If Base is not specified, the base defaults to 10. Returns #NUM! if either the given Number is less than or equal to 0 or Base is less than or equal to 1. Returns #VALUE! if the given Number or Base is not a number.
LOG10 (Number) (Link)	Returns the logarithm of the given Number using base 10. Returns #NUM! if the given Number is less than or equal to 0. Returns #VALUE! if the given Number is not a number.

It is strongly recommended to use the [GetCorrespondingMatch](#) function which is highly optimized and has performance 90% faster than MATCH.

Searches an array for **Number** using **MatchType**. Returns the one-based index of the first cell in the array/range that contains the value. If no match is found, returns #N/A.

MATCH(Number, Range, [MatchType])
[\(Link\)](#)

MatchType: This is optional. If omitted, the default value is 0. It can be one of the following:

- -1 = Returns the index of the largest number in the array/range that is less than or equal to **Number**. Array/range must be in descending order.
- 0 = Returns the index of the first number in the array/range that is equal to **Number**.
- 1 = Returns the index of the smallest number in the array/range that is greater than or equal to **Number**. Array/range must be in ascending order.

<p>MAX(Numbers, [...]) (Link)</p>	<p>Returns the number of the maximum value out of the given Numbers. Returns #VALUE! if one of the Numbers could not be interpreted as a number.</p>
<p>MAXL(Value, Numbers, [...]) (Link)</p>	<p>Returns the number of the maximum value out of the given Numbers that is greater than or equal to the given Value. Returns #VALUE! if the given Value could not be interpreted as a number or if one of the given Numbers could not be interpreted as a number.</p>
<p>MEDIAN(Numbers, [...]) (Link)</p>	<p>Returns the median value of the given Numbers. Null values are not counted as part of the median. No value is returned if there are no numbers. Returns #VALUE! if one of the Numbers given could not be interpreted as a number.</p>

<p>MID(Text, Offset, Count) (Link)</p>	<p>Returns the text string middle of the given string starting at the given Offset including the given number of characters. If Offset is 0, the result will start at the beginning of the string. If Count is greater than the length of the given Text string, the returned text will go up to the end of the given Text string. If Offset is 0, or Offset is greater than the length of the given Text string, an empty string will be returned. Returns #NUM! if Offset or Count is not an integer value. Returns #VALUE! if Text is not a text value, or if Offset could not be interpreted as an integer value.</p>
<p>MILLISECOND(Serial DateTime Value) (Link)</p>	<p>Returns the milliseconds of the given Serial DateTime value. This function will return the range 0-999. Returns #VALUE! if the given value cannot be interpreted as a date-time value.</p>
<p>MIN(Numbers, [...]) (Link)</p>	<p>Returns the number of the minimum value out of the given Numbers. Returns #VALUE! if one of the Numbers could not be interpreted as a number.</p>

MING (Value, Numbers, [...]) (Link)	Returns the number of minimum value out of given Numbers that are less than the given Value. Returns #VALUE! if the given Value is not one of the given Numbers. Numbers not be interpreted as text.
MINUTE (Serial DateTime Value) (Link)	Returns the minute for the given Serial DateTime Value. The function will return a value in the range 0-59. Returns #VALUE! if the given DateTime value is not interpreted as a serial date-time value.
MINZ (Numbers, [...]) (Link)	Returns the number of minimum value out of given Numbers that are greater than zero. Returns #VALUE! if one of the given Numbers is not interpreted as a number.
MOD (Number, Divisor) (Link)	The remainder of Number divided by Divisor. Returns #DIV/0! if given Divisor is 0. Can also return value using the formula: $\text{Number} - \text{INT}(\text{Number}/\text{Divisor}) \times \text{Divisor}$. The result when the numbers are not integers.

<p>MODE(Values, [...]) (Link)</p>	<p>Returns the value most frequently among Values. Returns the first value in the order given if different values are most frequent. Returns #N/A if the values that occur only once. Empty values are not counted. This function works with both numeric and text values. If there is an error value among the given values, the error value is returned.</p>
<p>MONTH(DateTime) (Link)</p>	<p>Returns the month for the given Serial DateTime Value. Returns the month returned for the first year of the given year (January = 1, December = 12). Returns #VALUE! if the given value cannot be interpreted as a date value.</p>
<p>MOSTRECENTNONZEROVALUE(Range) (Link)</p>	<p>Returns the first non-zero value in the given Range. Values of zero are considered equal to zero. Returns a null value if all the values in the Range are either null or zero.</p>

<p>MROUND(Number, Multiple) (Link)</p>	<p>Rounds the given Number to the nearest number that is a multiple of the given Multiple. If the given Number does not already satisfy this condition. If the given Number is rounded to the nearest whole integer, the result will be an integer, otherwise it will be a double. Returns an error if the given Multiple is negative. Returns an error if either the given Number or the given Multiple is not a number.</p>
<p>MROUNDDOWN() (Link)</p>	<p>Refer to FLOOR.</p>
<p>MROUNDUP() (Link)</p>	<p>Refer to CEILING.</p>
<p>NETWORKDAYS(StartDate, EndDate, [Holidays]) (Link)</p>	<p>Returns work days between the given StartDate and EndDate, including both the StartDate and the EndDate, not counting weekends or the Holidays.</p>

NMATCH(N, Value, Range, [MatchType])
([Link](#))

Same as **MATCH**, search continues until a matching item is found. If N is a positive integer, the given **Range** for **Value** using **MatchType** comparison method returns the one-based index into **Range** indicating the position of the match. Returns an error if no matches were found.

MatchType: This is the comparison method. The default value is 0. One of the following:

- -1 = Returns the index of the first value in the given **Range** that is greater than or equal to the given **Value**. Values in **Range** must be in ascending order.
- 0 = Returns the index of the first value in the given **Range** that is equal to the given **Value**.
- 1 = Returns the index of the first value in the given **Range** that is less than or equal to the given **Value**. Values in **Range** must be in ascending order.

<p>NORM.DIST(X, Mean, StdDev, Cumulative) (Link)</p>	<p>Computes the Normal Distribution for X, Mean, StdDev. If Cumulative is set to 1, the function returns the Cumulative Distribution Function (CDF). If Cumulative is set to 0, the function returns the Probability Distribution Function (PDF).</p>
<p>NORM.S.DIST(X, Cumulative) (Link)</p>	<p>Computes the Standard Normal Distribution for X (with Mean 0 and StdDev 1). If Cumulative is set to 1, the function returns the Cumulative Distribution Function (CDF). If Cumulative is set to 0, the function returns the Probability Distribution Function (PDF).</p>
<p>NOW() (Link)</p>	<p>Returns the current date and time as a Serial Value. The serial date system represents 2.0 as midnight, January 1, 1900, and 2.5 represents noon, January 1, 1900.</p>
<p>ODD(Number) (Link)</p>	<p>If the number is not an odd number, it is rounded up to the next odd number. If the Number is less than 0, it is rounded away from zero. #VALUE! if the argument is not a number.</p>

OFFSET (From, Rows, Columns, [Height], [Width]) (Link)	Returns the reference range calculated from the reference or a cell range as the first argument and the offset values.
OR (Boolean, [...]) (Link)	Returns FALSE if all of the given parameters are equal to FALSE; otherwise, it returns TRUE. Returns an error if one of the given parameters could not be interpreted as a boolean value. Examples: =OR(E3 < 10, E3 > 20) (Spreadsheet Study for) =OR(SG1 > 40, SG1 < 30) (Simple Alert formula) =OR(AND(H > 100, L < 80), AND(L < 80, SG1 < 30)) (Simple Alert formula)
PEARSON (Array1, Array2) (Link)	Returns the Pearson product-moment correlation coefficient between two arrays, which must be between -1.0 and 1.0. Returns #VALUE! if the size of the arrays are not equal or the arrays are empty. Returns #DIV/0! if the standard deviation of either array is zero.

PERCENTILE(Numbers, K) [\(Link\)](#)

Returns the K-th percentile values in a cell range. If linear interpolation values if K is not a multiple of (n - 1), where n is the number of numerical values in the cell range. Returns #VALUE! if K is not a number. Returns #NUM! if K is less than 0 or greater than 1.

PERCENTRANK(Array, X, [Significance])
[\(Link\)](#)

Returns the rank of a value in a data set as a percentage of the data set size. If the value is found in between two values in the set, the return value is calculated by linear interpolation. Returns #VALUE! if X is less than the smallest value or greater than the biggest value. Returns #NUM! if the array is empty. If the third argument is provided it will control the number of digits will be used to display the values. If the third argument is less than 1 the #NUM! error will be returned.

PROB(Values, Probability, LowerLimit, [UpperLimit]) ([Link](#))

Calculates the probability values in a range against two limits. If the upper limit is omitted calculates the probability that values in the range are greater than or equal to the lower limit. If the lower limit is omitted and the Probability argument is omitted, the function calculates the probability that values in the range are less than or equal to the lower limit. If the total of the probabilities is not 1.0 the function returns a #NUM! error.

RADIANS(Degrees) ([Link](#))

Converts the given Degrees to radians. Returns #VALUE! if Degrees is not a number.

RAND() Or **RAND**(Low, High) ([Link](#))

Generates a random number. If no arguments are given, returns a random number from 0.0 to 1.0. If one argument is given, returns a random number from 0.0 to the argument. If two arguments are given, returns a random decimal number between Low and High. Return values are greater than or equal to Low and less than or equal to High. Returns #VALUE! if the arguments are not numbers or High could not be greater than Low.

RANDINT() Or **RANDINT(Low, High)** [\(Link\)](#)

Generates a random number. If the single Count is given, randomly selected integer to Count. Returns Count is less than 1 arguments Low and given, returns a selected integer from High. Returns #NUM! if Count, Low, or High is not an integer value.

RIGHT(Text, Count) [\(Link\)](#)

Returns the given Count of characters from the right of the given Text string. If Count is greater than the length of the given Text string, the string will be returned in its entirety. If Count is 0, an empty string is returned. If the given Count is negative, the entire length of the string, except for the specified number of characters, will be returned. Returns #VALUE! if Text is not a string, or Count is not a number.

ROUND(Number, [Digits = 0]) [\(Link\)](#)

Rounds the given Number to the nearest number with the specified number of digits. If Digits is not given, the given Number is rounded to the nearest integer. If Digits is positive, the given Number is rounded to the nearest number with the specified number of decimal digits. If Digits is negative, the given Number is rounded to the nearest multiple of 10^{-Digits} that is a multiple of 10. Example: ROUND(1.235,1) = 1.2; ROUND(1.235,2) = 1.24; ROUND(1235,-1) = 1240. If Digits >= 0, the return value will be an integer, otherwise it will be a double. Returns the given Number if Digits is not given, or if the given Number (when given) is not a number.

ROUNDDOWN(Number, [Digits = 0]) ([Link](#))

Rounds the given Number to the next number of given Digits. If Digits is 0, the given Number does not change. If Digits is negative, rounding down means closer to zero. If Digits is 0 and the given Number is positive, the rounded number will be less than or equal to the given Number, and if the given Number is negative, the rounded number will be greater than or equal to the given Number. If Digits is positive, the given Number is rounded down to the next integer. If Digits is 0 and the given Number is rounded to the next number, the result will be many decimal digits. Examples: `ROUND(1.235, 2)` returns 1.24, `ROUND(1.235, 3)` returns 1.235. If Digits is negative, the given Number is rounded to the next number that is a power of 10. Examples: `ROUND(1235, -1)` returns 1000, `ROUND(1235, -2)` returns 100. If Digits ≥ 0 , the result will be an integer, otherwise it will be a double. Returns the given Number if Digits is 0, or if the given Number (when given) is not an integer.

ROUNDUP(Number, [Digits = 0])

(Link)

Rounds the given Number to the next number with the given Digits, if the Number does not already satisfy this condition. Rounding means farther away from zero for a positive number rounded up, and closer to zero for a negative number rounded up. If Digits is 0, the given Number is rounded up to the next integer. If Digits is positive, the given Number is rounded up to the next number with the given decimal digits. If Digits is negative, the Number is rounded up to the number that is a multiple of 10^{-Digits}.
ROUNDUP(1.235,1)
ROUNDUP(1.235,2)
 If Digits is negative, the Number is rounded up to the number that is a multiple of 10^{-Digits}.
ROUNDUP(1235, -1)
ROUNDUP(1235, -2)
 If Digits >= 0, the return value will be an integer, otherwise it will be a double. Returns the given Number if the given number, or if the given Digits (when given) is not a number.

ROW([Reference]) [\(Link\)](#)

Returns the absolute (not relative) number (not the row of the given Formula Reference) if the given Formula Reference is not given. The formula is returned as #VALUE! if Reference is not a cell reference type. The absolute number of the first row (1) is 1.

ROWS(Range) [\(Link\)](#)

Returns the number of rows in the given Range. If the given Range is not a cell range reference type, it returns #VALUE!.

SEARCH(SubString, FullString,
[StartPosition = 1]) [\(Link\)](#)

Returns the starting position of the first instance of SubString text within FullString text. The StartPosition parameter is 1-based, meaning the position of the first character of SubString is found at 1. If the StartPosition is greater than the length of FullString, the return value will be 0. By default, SEARCH performs a case-insensitive search for the given SubString starting at the beginning of FullString, but this can be changed by specifying a value for StartPosition. The search is case-insensitive by default. [FIND](#) for a case-sensitive search). Returns 1 if SubString is empty. Returns #VALUE! if no match is found. Returns the given StartPosition if the given StartPosition is greater than the length of the given FullString or one of the strings is empty. Returns the given value, or the given StartPosition if the given StartPosition is not an integer value. Returns #NUM! if the given StartPosition is less than 1.

SECOND([Serial DateTime Value](#)) [\(Link\)](#)

Returns the second of the month for the given [Serial DateTime Value](#). The function will return a value in the range 0-59. Returns the given DateTime value interpreted as a serial value.

SIGN (Number) (Link)	Returns 1 if the given Number is positive, -1 if the given Number is negative, or 0 if the given Number is zero. Returns #VALUE! if the given Number is not a number.
SIN (Number) (Link)	Returns the sine of the given Number. Returns #VALUE! if the given Number is not a number.
SINH (Number) (Link)	Returns the hyperbolic sine of the given Number. Returns #VALUE! if the given Number is not a number.

SLOPE(KnownYs, KnownXs) [\(Link\)](#)

Returns the slope of the line through a set of known X and Y values. The first parameter is a reference to a range of cells containing the known Y values, and the second parameter is a reference to a range of cells containing the known X values. Both ranges will be the same size. Returns #NUM! if the number of known Y values does not match the number of known X values, or the slope is vertical.

Example:

=SLOPE(E3:E12, A3:A12)

This formula will calculate the slope of the line through the data points in the range E3:E12, A3:A12. The result will be a positive value if the data is generally ascending and a negative value if the data is generally descending.

SMALL(Numbers, NthSmallest) [\(Link\)](#)

Returns the Nth smallest value in an array. The first argument is the array of values, and the second argument is the number of the smallest value you want. If the second argument is 1, it returns the smallest value. If it is 2, it returns the second smallest value, and so on. Empty cells are ignored. If the second argument is less than 1 or greater than the number of values in the array, it returns #NUM!.

SQRT (Numbers) (Link)	The square root of Number. Returns #N/A if given Number is less than 0. Returns #VALUE! if Number is not a number.
STANDARDIZE (X, Mean, StandardDev) (Link)	Returns the normalized value from a distribution characterized by the given Mean and StandardDev. The normalized value is (X - Mean) / StandardDev. Returns #NUM! if the given StandardDev is less than or equal to 0.
STDEV (Numbers, [...]) (Link)	Refer to STDEV.S .
STDEV.P (Numbers, [...]) (Link)	Returns the standard deviation of a population. Empty cells are ignored. Returns #VALUE! if there are fewer than 2 numbers. If the Numbers are interpreted as a range, there must be at least one number in the range.
STDEV.S (Numbers, [...]) (Link)	Returns the standard deviation of a sample. Empty cells are ignored. Returns #VALUE! if there are fewer than 2 numbers. If the Numbers are interpreted as a range, there must be at least one number in the range. For more information, refer to STDEV .

<p>SUM(Numbers, [...]) (Link)</p>	<p>The total of all the numbers added together. If all Numbers are integers the result will be an integer. Otherwise the result will be a double value. Returns #VALUE! if one of the given arguments could not be interpreted as a number.</p>
<p>SUMPRODUCT(Range1, [Range2, ...]) (Link)</p>	<p>Multiplies the corresponding values in all of the given ranges and then returns the sum of those products. For example, if three ranges are given, the values of the first cells in each range will be multiplied together, that product will be added to the product of the second three ranges, and so on until the last cell of all three ranges. Cells that do not contain a numeric value are treated as having a value of zero. Returns #VALUE! if any of the arguments are not cell ranges. Returns #NUM! if all arguments do not have the same number of cells.</p>
<p>TAN(Numbers) (Link)</p>	<p>The tangent of the given number. Returns #VALUE! if the number is not a number.</p>

TANH(Numbers) [\(Link\)](#)

The hyperbolic tangent of the given number. Returns the given number if the given number is not a number.

Returns the given text value. The given text value is used for formatting the text value. It is optional and has no effect on other values.

To format a number to a specific number of decimal places, use a format string. For example, ".000" or ".###". For example, following the decimal format, that many decimal places will be displayed, even if they are insignificant. For example, TEXT(0.05, "0.000") returns "0.050". For each number (#) following the decimal format, that many decimal points will be displayed for trailing zeros. For example, TEXT(0.05, "0.###") returns "0.05".

TEXT(Value, Format) [\(Link\)](#)

These two methods are combined such that they can format numbers with a maximum of four decimal places. Values will be rounded to the maximum number of digits that can be shown.

Zeros placed before the decimal point indicate to the left of the decimal point.

zeros for the whole r
example: TEXT(2.5,
will return "002.5".

Number signs (#)
decimal point have no
decimal point in the
must match the gl
that specifies the de
character. The dec
character used in tl
value will match
setting.

TIME(Hour, Minute, Second, [Millisecond = 0]) [\(Link\)](#)

Returns a [Serial Date](#) for the given Hour, Minute, and Millisecond. If Millisecond is not given or 0 is used for the Millisecond portion. Returns #N/A if the given Hour, Minute, or Second does not represent a valid time. Returns #N/A if any of the parameters are not interpreted as integers.

Here is an example of how to use the TIME function in one of the columns at Sheet 1. The formula returns TRUE(1) when the value in the Date-Time column of the Sheet used in the Spreadsheet Study is greater than the times 9:30:00 and 9:34:59. The formula is: **=AND(FRACETIME(A3, 29, 59, 750), FRACETIME(A3, 29, 59, 750) < TIME(9, 34, 59, 250))**

Milliseconds are used to provide this an accurate comparison and to avoid floating-point error.

TIMEVALUE(Text) [\(Link\)](#)

Returns a [Serial Date](#) for the given Text, interpreted as a time string. Returns #N/A if the given Text cannot be interpreted as a time value. Returns #VALUE! if Text cannot be interpreted as a text value.

TODAY() [\(Link\)](#)

Returns the current local time as a [Serial DateTime Value](#). The date value is the number of days since 1899-12-30 (the date that Excel started using the serial date value for 1900 is 2).

TRIMMEAN(Range, Percent) [\(Link\)](#)

Sorts the values in Range, and returns the average of the inner values, excluding a percentage of the outer values (the extremes). The number of values excluded is the number of values in Range multiplied by the given Percent, rounded down to the nearest multiple of 2. This way, an even number of values are excluded from both the high and the low end of the range of values. Cells with no values are not counted as part of the range. Return #DIV/0! if there are either no values or the average, or the given Percent is less than zero or greater than one. Returns #VALUE! if the given Range contains non-numeric values.

TRUNC(Number, [Digits = 0]) ([Link](#))

Returns the given number truncated to the given number of Digits. If the given Digits is 0 or not specified, the Number is truncated to a whole number. If the number of Digits is greater than 0, then the given Number is truncated to that many digits. If the given Digits is less than 0, the given Number is truncated to a multiple of $10^{-(\text{Digits})}$. Returns #VALUE! if the given Digits is not a number, or the given Number is not an integer.

TRUNC HOUR(DateTime, [Hours = 1]) ([Link](#))

Returns the given DateTime truncated down to the interval of Hours. If Hours is greater than 0, the given DateTime is truncated down to the interval of Hours (e.g. 2 hours, 3 hours, etc.). Returns #NUM! if Hours is less than 0. Returns #VALUE! if the given Hours is not a number, or the given DateTime is not an integer.

<p>TRUNCMIN(DateTime, [Minutes = 1]) (Link)</p>	<p>Returns the given truncated down to minute. If Minutes is 1, the given DateTime is truncated down to the given Minutes minutes, 10 minutes. Returns #NUM! if Minutes is less than 1, or #VALUE! if the given DateTime is not a number, or Minutes is not an integer.</p>
<p>TRUNCSEC(DateTime, [Seconds = 1]) (Link)</p>	<p>Returns the given truncated down to second. If Seconds is 1, the given DateTime is truncated down to the given Seconds seconds, 10 seconds. Returns #NUM! if Seconds is less than 1, or #VALUE! if the given DateTime is not a number, or Seconds is not an integer.</p>
<p>TYPE(Argument) (Link)</p>	<p>Returns the type of a value encoded as an integer. Cells considered to contain a numerical value return the following values:</p> <ul style="list-style-type: none"> 1 = Number 2 = Text string 4 = Logical value 16 = Error value 64 = Cell range

VALUE(Text) [\(Link\)](#)

Converts the given number. Returns #VALUE! if text cannot be converted into a numeric value type the Text, that value returned.

VLOOKUP(Value, Range, ColumnInRange, [ApproximateMatch]) [\(Link\)](#)

Searches for the given Value in the left-most column of the given **Range**, and returns the value of the cell at the same row as the found value, at the given **ColumnInRange**.

ColumnInRange is the column index within the given **Range**, where 1 is the left-most column of the given **Range**.

If **ApproximateMatch** is given as **TRUE**, and no exact match is found, the last value less than or equal to the given **Value** will be used.

When **ApproximateMatch** is given as **TRUE**, the values in the left-most column of **Range** are assumed to be in ascending order.

If **ApproximateMatch** is given or given as **FALSE**, and no exact match for the given **Value** is found, #VALUE! is returned.

Returns #VALUE! if the given **Value** cannot be resolved.

	<p>actual value (such as a date or reference is given), ColumnInRange is interpreted as an integer. If the given Approximate cannot be interpreted as a boolean value.</p> <p>Returns #REF! if Range is not valid, or ColumnInRange is not a number of columns in Range.</p> <p>Returns #NUM! if ColumnInRange is less than 1.</p>
WEEKDAY (Serial DateTime Value) (Link)	<p>Returns an integer representing the day of the week for the given Serial DateTime Value. The return value will be in the range of 1-7 (1=Sunday, 7=Saturday). Returns #VALUE! if DateTime cannot be interpreted as a serial date value.</p>
WEEKNUM (Serial DateTime Value) (Link)	<p>Returns an integer representing the week number of the year for the given Serial DateTime Value. The return value will be in the range of 1-53. A date of the 1st of any given year will return a value of 1. Weeks start at Monday. Returns #VALUE! if the given DateTime cannot be interpreted as a serial date value.</p>

<p>WEIGHTEDMOVINGAVERAGE(Range) (Link)</p>	<p>Returns the Weighted Average of the inputted values. Returns #VALUE! if given range of data contain numeric values. Moving Average - VBA information on the calculation.</p>
<p>WORKDAY(StartDate, WorkDays, [Holidays]) (Link)</p>	<p>Returns the serial date given number of days before or after StartDate. Work days exclude weekends and Holidays.</p>
<p>YEAR(DateTime) (Link)</p>	<p>Returns the year for Serial Date Time Value. Returns #VALUE! if DateTime cannot be as a serial date value.</p>
<p>ZTEST(Array, Mu, [Sigma]) (Link)</p>	<p>Calculates the one-tailed test for comparing the sample mean of the Array to the hypothesized population mean. Mu argument is the hypothesized population mean, standard deviation, omitted, then the sample standard deviation will be used instead. If Sigma is empty the #N/A! error is returned. A two-tailed test is done as $2 * \text{MIN}(\text{ZTEST}(\text{Array}, \text{Mu}, [\text{Sigma}]), 1 - \text{ZTEST}(\text{Array}, \text{Mu}, [\text{Sigma}]))$.</p>

*Last modified Wednesday, 22nd February, 2023.